CALCULUS, 2016-17-Q1

Bioinformatics degree

LAB 1. Maths with `Python`*: an introduction.*

In this practice, we will mostly cover basic computations (some of them from class examples), emphasizing on the numerical aspects and the associated procedures via `Python`.

You can find basic information about `Python` for mathematics in the following links:

- Basic tools: https://docs.python.org/2/library/numeric.html within https://docs.python.org/2/library/index.html.
- `NumPy`: https://docs.scipy.org/doc/numpy/index.html. The fundamental package needed for scientific computing with `Python`. It contains a powerful N-dimensional array object, sophisticated (broadcasting) functions, basic linear algebra functions, sophisticated random number capabilities, tools for integrating Fortran and C/C++ code,...
- `matplotlib.pyplot`: http://matplotlib.org/users/pyplot_tutorial.html. Collection of command style functions that make `matplotlib` work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Other useful remarks:

- Vectors indexed from 0, as in C. Subscripts of vectors: v[j].
- Be careful with the tabulation of your editor. Instead of using Tab, use 4 simple spaces.
- Command range or np.arange does not consider the endpoint by default; range(3,7)=array(3,4,5,6). Use np.linspace for a more accurate treatment of endpoints.
- If you are using `Python 2`, remind the dots after numbers to indicate floating-point arithmetics.

Calculus Bioinformatics 16-17Q1

1. Comment what happens when typing these commands. You can download the file `Lab0000-Preliminaries.py` to avoid typing.

```
x= 1.e-15
((1+x)-1)/x
```

   What should be the answer? Try with other values of x. For instance, 0.1, 0.001, etc and look for an explanation.

2. Execute the following loop (write on an editor, taking care of leaving 4 spaces at the beginning of the last line):

```
a= 1.
b= 1.
while a+b!=a:
    b= b/2.
```

   According to our understanding, the loop should never finish, since b is a strictly positive number. But the loop ends!

   Which is the value of b? Find an explanation.

3. `Python` has facilities to find the real and complex roots of polynomials. We need a specific package from `NumPy`. For instance, we define the polynomial $p(x) = 5x^3 + 3x^2 - x + 2$ by typing

```
from numpy.polynomial import Polynomial as P
p = P([5, 3, -1, 2])
```

   and the command

```
p.roots()
```

   provides the three roots of the polynomial. Play with other examples.

   You may find interesting the following example, corresponding to a polynomial of degree 7, in fact $(x+1)^7$.

```
p= P([1, 7, 21, 35, 35, 21, 7, 1])
```

   Find the roots obtained by `Python`.

4. In Problem 6 (List 1), we studied a relation between the fraction d of individuals who contract the disease in terms of the basic reproductive number R_0:

$$f(d; R_0) := R_0\, d + \log(1-d) = 0.$$

   For the ebola virus disease, for instance, the basic reproductive number has been estimated to belong to the interval $I_E := [1.5, 2.5]$.

Calculus Bioinformatics 1617Q1

(a) Use the code from `Lab0020-Zeros.py` to plot 6 different curves $\{f(d; R_0), d \in [0, 1]\}$, with R_0 varying in I_E.

(b) Visualize the zeroes of f for this set of R_0 values and adapt the code from `Lab0020-Zeros.py` to find an interval of the fraction of individuals who contract the ebola virus.

(c) Put R_0 in terms of d and plot (adapting the code from `Lab0020-Zeros.py`) the fraction of individuals versus the basic reproductive number for $R_0 \in I_E$. Observe that you obtain a graph of the prediction about the ebola virus disease incidence.

5. In Lecture01, we treated the modeling of drug clearance (see also Example 1.1.1 in [Ledder]) with data obtained from a hypothetical clinical analysis in order to test the effects of two tablets of acetaminophen in one subject's body. We assume here that the results of the analysis are those shown in the following table:

Time (h)	0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0
Acetaminophen (mg)	650.0	552.3	471.5	406.7	347.1	297.6	254.9	219.8	184.2

We aim at testing the exponential decay behavior

$$y(t) = e^{-\kappa t} A \tag{1}$$

of the drug clearance in the subject's body.

(a) Use the code from `Lab0010-ExponentialDecay.py` to plot the points of the table in the space "Time-Acetaminophen" $t - y$.

(b) Apply logarithms to make the expression (1) linear in terms of t.

(c) Use the code from `Lab0010-ExponentialDecay.py` to plot the points of the table in the space $t - \log y$. Write the corresponding output into a file called "FigureLog.png".

(d) Assuming $A = 650.0$ mg (initial value), we aim at minimizing the *residual sum of squares*:

$$RSS(\kappa) := \sum_{i=0}^{8} \left(\log(A/y_i) - \kappa t_i\right)^2,$$

where (t_i, y_i) for $i = 0, \ldots 8$ are the values of the table, in terms of κ. Prove that the minimum must satisfy

$$\kappa = \frac{\displaystyle\sum_{i=0}^{8} t_i \log(A/y_i)}{\displaystyle\sum_{i=0}^{8} t_i^2}.$$

(e) Use the code from `Lab0010-ExponentialDecay.py` to compute the optimal κ for the given data set. Plot the points of the table and the best-fitting curve both for in the $t - \log y$ and the $t - y$ plane.

Calculus Bioinformatics 161701

CALCULUS, 2016-17-Q1, Bioinformatics degree

LAB 2. Derivatives, Interpolation and Taylor expansions.

1. We consider the additional difficulties of modeling ibuprofen amounts in a patient, given that ibuprofen is slowly absorbed from the digestive system and relatively quickly eliminated from the body.

   (a) Plot a concentration curve for ibuprofen using the model $y = A\,e^{-k\,t}$. Assume a dose of 200 mg and a rate constant of $k = 0.35$. Hint: you can adapt either the code from `Lab0010-ExponentialDecay.py` or from `Lab0020-Zeros.py` to plot the function.

   (b) A more sophisticated drug clearance model takes account of the need for an oral dose to be absorbed by the digestive system. This model is

   $$y(t) = \frac{A\,b}{b-k}\left(e^{-k\,t} - e^{-b\,t}\right),$$

   where A is the amount of the dose, k is the clearance rate for the bloodstream, and b is the clearance rate for the digestive system. Normally, $b > k$. Use this model to plot a drug versus time curve for a 200 mg dose of ibuprofen, assuming $k = 0.35$ and $b = 0.46$.

   (c) Describe the important differences between the curves of part (1a) and part (1b). Consider qualitative features of the graphs as well as quantitative properties ( )[1] such as the peak value and the time at which the corresponding peaks occur. Discuss, both analytically and numerically, the role of parameter b.

2. In order to obtain information about an individual's glucose tolerance, recordings of glucose levels are performed in clinical practice to control diabetes. It is common practice to use simple summary measures, such as fasting[2] value, two-hour value or the area under the curve. Table shows OGTTs (Oral Glucose Tolerance Tests) with glucose measurements over two hours recorded for healthy pregnant women in their first trimester (averages over population), after ingesting a 75 g preparation of glucose.

t (min)	0	10	20	30	40	50	60	70	80	90	100	110	120
x (mmol/l)	4.00	5.02	5.56	5.77	5.75	5.56	5.27	4.91	4.52	4.12	3.73	3.34	2.98

   (a) Adapt the code in `Lab0040-Interpolation.py` to compute the interpolating polynomial of the table subset containing only the measurements at $t = 0,\ 30,\ 60,\ 90$ and 120 minutes.

   (b) Give a prediction of the glucose levels at $t = 40$ and $t = 50$ and compare the result with the actual values given in the full table. If it was the case that the table had a perfect fit with a hypothetical smooth function f, could you provide an upper bound for $|f^{(v)}(x)|$ in $[0, 120]$?

[1] We use the symbol to indicate that some part of a problem requires computations by hand.
[2] dejú (cat), ayuno (cast), jeûne (fr)

Calculus Bioinformatics 16-17Q1

(c) Adapt the code in `Lab0040-Interpolation.py` to compute both the interpolating polynomial $P_{12}(t)$ and the interpolating spline $S_{12}(t)$ of the full table and compare them, both visually and numerically (you can compute $\frac{1}{M}\sum_{j=0}^{M}|P_{12}(t_j) - S_{12}(t_j)|$, where $t_j = 120\,j/M$ and M is a "sufficiently" large integer).

3. Population dynamics is the branch of life sciences that studies the size and age composition of populations as dynamical systems, and the biological and environmental processes driving them (such as birth and death rates, migrations, ...). A simple but already realistic model is the differential equation including logistic growth with emigration:

$$\dot{x} = r\,x\,(1 - K\,x) - c, \qquad r > 0, K > 0, c > 0, \tag{1}$$

where $x = x(t)$ stands for the size of a biological population at time t, $\dot{x} = \frac{d}{dt}x(t)$, r is the birth rate, K the carrying capacity in absence of emigration (e.g., due to the competition for food) and c models the rate of emigration. In fact, the model does not only model emigration but also other subtracting effects like harvesting, hunting or fishing on a population with logistic growth. Our aim is to study these effects through the control parameter c.

(a) A value of x such that $r\,x\,(1 - K\,x) - c = 0$ is called *equilibrium of the system*. Give an expression of the equilibria of system (1) in terms of the parameters r, K and c. In particular, prove that $c_* = \dfrac{r}{4\,K}$ is a *bifurcation value*, at which the system changes its qualitative behaviour.

(b) For $r = 0.5$ and $K = 20$, use the code in `Lab0030-1DBifurcationDiagram.py` to plot the equilibria versus the parameter c and observe the effects of a high emigration or harvesting.

4. Consider a function f which is differentiable up to order at least 2 at the point x_0.

(a) Using Taylor expansions, prove the validity of the following approximations and expand the right-hand sides in terms of h:

- $f'(x_0) \sim \dfrac{f(x_0 + h) - f(x_0)}{h}$ (lateral/forward approximation of the 1st derivative);
- $f'(x_0) \sim \dfrac{f(x_0 + h) - f(x_0 - h)}{2\,h}$ (central approximation of the 1st derivative);
- $f''(x_0) \sim \dfrac{f(x_0 + h) - 2\,f(x_0) + f(x_0 - h)}{h^2}$ (central approximation of the second derivative).

Note that the order in h of the the right-hand sides expressions is directly related with the accuracy of these numerical approximations.

(b) Consider $f(x) = \ln(1 + x)$; then, using the formulas in (4a) and adapting the first part of the code in `Lab0050-SymbolicDerivatives.py`, compute $f'(0.5)$ and $f''(0.5)$ for different values of h (for instance, $h = 10^{-j}$ with $j \geq 1$). Assess the validity of the predicted errors.

(c) Despite of these numerical approximations, derivatives of elementary functions can be easily computed from derivation rules. Using symbolic mathematics (also known as *computer algebra systems*), one can implement the derivatives of the basic elementary functions together with the derivation rules. In `Python`, there exists a library called `SymPy`[3] that can do this job. Compute again the derivatives from (4b) using or adapting the last three lines of the script provided in `Lab0050-SymbolicDerivatives.py`

[3] See http://www.sympy.org/en/index.html.

Calculus Bioinformatics 1617Q1

5. Consider the function $f(x) = \sin(\ln(1+x))$ and examine the functions programmed in `Lab0060-Taylor.py`.

   (a) Find the expression of the Taylor polynomials $P_1(x)$, $P_4(x)$, $P_7(x)$ and $P_{10}(x)$ at $x_0 = 0$, and plot them simultaneously.

   (b) Use $P_2(0.4)$ to approximate $f(0.4)$. Find and error bound for $|f(0.4) - P_2(0.4)|$ using the error formula ( ) and compare it with the actual error.

   (c) How many terms should we add to compute $f(0.4)$ with a maximum error of 10^{-8}?

Calculus Bioinformatics 1617Q1